Office de la Propriété Intellectuelle du Canada
Un organisme d'Industrie Canada

Canadian Intellectual Property Office
An agency of Industry Canada

CA 2303466 A1 2001/09/30

(21) 2 303 466

(12) DEMANDE DE BREVET CANADIEN
CANADIAN PATENT APPLICATION

(13) A1

(22) Date de dépôt/Filing Date: 2000/03/30

(41) Mise à la disp. pub./Open to Public Insp.: 2001/09/30

(51) Cl.Int.$^7$/Int.Cl.$^7$ G06F 17/30, G06F 17/60

(71) Demandeur/Applicant:
SOFTV.NET, CA

(72) Inventeurs/Inventors:
LOEN, RICHARD, CA;
ABRAHAM, STEPHEN, CA

(74) Agent: GOWLING LAFLEUR HENDERSON LLP

(54) Titre : PAGES WEB BASEES SUR UN GABARIT A MISE A JOUR AUTOMATIQUE AVEC EVENEMENTS DECLENCHEURS ET PLANIFICATION

(54) Title: AUTOMATICALLY UPDATING TEMPLATE BASED WEB PAGES WITH TRIGGER EVENTS AND SCHEDULING

(57) Abrégé/Abstract:

A template based approach is provided for creating web pages. The template uses regions which are identified by the web page developer. Each region can be either text, graphics, video or date/time. Regions can include audio, weather, news, advertising, sports or financial information. The web page regions are is automatically updated when the source file is updated without having to republish or refresh the web page. Text files can also automatically page through multiple pages when the file is larger than the specified text region. The web page can further execute Event Triggers for executing actions based on events. Further, the web page can include scheduling information for executing actions based on time or events. The invention uses graphical user interfaces to eliminate the need for complex programming to create web pages incorporating the above mentioned features. When a web page is published to the Internet, a software module generates the required code and uploads the data to a viewer's computer in HTML format.

## Abstract

A template based approach is provided for creating web pages. The template uses regions which are identified by the web page devcloper. Each region can be either text, graphics, video or date/time. Regions can include audio, weather, news, advertising, sports or financial information. The web page regions are is automatically updated when the source file is updated without having to republish or refresh the web page. Text files can also automatically page through multiple pages when the file is larger than the specified text region. The web page can further execute Event Triggers for executing actions based on events. Further, the web page can include scheduling information for executing actions based on time or events. The invention uses graphical user interfaces to eliminate the need for complex programming to create web pages incorporating the above mentioned features. When a web page is published to the Internet, a software module generates the required code and uploads the data to a viewer's computer in HTML format.

# AUTOMATICALLY UPDATING TEMPLATE BASED WEB PAGES WITH TRIGGER EVENTS AND SCHEDULING

## Field of the Invention

The invention relates generally to web page publishing, and more particularly to a method of creating a web page using a software application interface.

## Background of the Invention

The technology of the Internet is well known in the art and the importance of the Internet is well known by the general public. The Internet is an ever-increasing tool in the world of business. As more and more people log on, the more demand there is for increasingly attractive and interactive sites. Thc reasons for wanting to produce a more appealing site are numerous and include: to improve each user's experience at the site, to create personalized content on the site, to generate public interest for advertising on the site.

Prior web page design involved tedious coding of HyperText Markup Language (HTML) codes using a text editor. It was necessary for a developer to learn a great deal of complicated code and to learn the differences in each type of web browser in order to build web pages. This often required companies to employ expensive programmers to build and maintain corporate web pages.

More recently, applications have been designed to allow web page developers to create web pages using a WYSIWYG (What You See Is What You Get) interface. This has allowed web developers to manipulate text and images through a graphical user interface (GUI) rather than manually editing complicated HTML codes. However, even with these advanced systems, it is difficult to maintain web pages up to date. In order to update the content of a web page, the web developer must republish the web page to the web server. Viewers will not see the updated changes until they reload the page or "Refresh" the page.

In addition, web pages can only be programmed to automatically update with complex custom Java programming.

A particular web page that includes video content may be viewed by a user for any length of time, for example, from fifteen minutes to any number of hours. If the content on the web page is static, it becomes stale and uninteresting to that user. Therefore, web page developers have had to learn the arcane commands required to manipulate Dynamic HTML (DHTML) elements, Java scripts and Visual Basic scripts and embed these commands with programmatic solutions. Such efforts as DHTML allow web content to change but do not automatically update content from its original source file.

Extensive coding in current web languages has enabled web browsers to handle many sorts of events for quite some time. For example, web pages can respond to Mouse Clicks, Mouse Enters and Mouse Exits as well as keyboard commands. Current web languages have no built-in support for events. For example, to "move" or "resize" video content, the developer must manually program the commands, rather than rely on built-in support for the commands.

It is currently also very difficult to link various elements of a web page together so that HTML content is linked to video content or video content is linked to time-of-day for instance. The concept of time-of-day scheduling of events is known in the art. Currently there are three methods of accomplishing time-of-day scheduling of events. The first is tedious hand-coding of the schedule into Java code. Hand coding involves using a software developer to hand craft the code required to cause an action to occur at a particular time. This is very difficult to accomplish. The second is to use the Synchronized Multimedia Integration Language (SMIL) language. The SMIL language is not a component of either the Microsoft Internet Explorer® or Netscape Navigator® Browsers. SMIL commands can only be executed in the RealNetworks Player. This requires the web page viewer to have previously downloaded and installed the RealPlayer. The third method is to use HTML+Time. The problem with this solution is that it can only be viewed with Microsoft Internet Explorer® Version 5.0 or higher. In any other browser on the market, HTML+Time commands will simply be ignored. Web developers are unable to satisfy a major portion of their audience.

With the growing importance and popularity of the Internet, there is a serious need for a simpler, easier method of creating and formatting web pages. There is also a need to provide easier methods of dynamically controlling the content of HTML, video and audio elements, dynamically moving and resizing web page elements, and causing actions to occur on a web page at a specific point in time. In addition, there is a need to achieve the aforementioned without the user having a knowledge in programming.

There is also a need to provide a way to create web pages compatible with both Microsoft Internet Explorer® and Netscape Navigator®.

Therefore, the present invention has been developed in order to meet these demands.

**Summary of the Invention**

The present invention allows web developers to create and format dynamic, automatically updating web pages without the requirements of programming. Web pages can be laid out using a GUI where regions are defined and their properties are assigned simply by clicking and dragging a mouse pointer.

According to one aspect of the invention, a web page is created with the use of a template. Regions are specified in the template. The template and regions can have a number of properties specified.

According to another aspect of the invention, each region is associated with content of various types. Regions can be text, graphics, video and date/time.

According to another aspect of the invention, content can include audio, weather, news, advertising, sports or financial information.

According to another aspect of the invention, a web page is automatically updated when the source file is updated. The web page updates the affected content without affecting the rest of the web page content.

According to another aspect of the invention, text regions in a web page can automatically be displayed in sequential pages one at a time in the display area.

According to another aspect of the invention, automatically paging text remains on screen according to the Paging Rate.

According to another aspect of the invention, event triggers are built in and automatically handled.

According to another aspect of the invention, the content in one region is synchronized with the content in another region.

According to another aspect of the invention, Event Triggers can begin the execution of a schedule.

According to another aspect of the invention, schedules are created using a graphical user interface.

According to another aspect of the invention, schedules can be defined in one of three time frames, including Local Time, GMT Time and Offset Time.

**Brief Description of the Drawings**

The invention will be described with reference to the accompanying drawings, wherein:

Figure 1 illustrates the operating environment in which the present invention works;

Figure 2 illustrates the browsing environment in which the invention works;

Figure 3 illustrates the method of creating a web page according to the present invention;

Figure 4 illustrates the template screen display of the present invention;

Figure 5 - 12 illustrate the screen displays for the template properties with different tabs selected;

Figure 13-20 illustrate the screen displays for the region properties with different tab selected;

Figure 21 illustrates a typical web page that can be created according to the present invention;

Figure 22 illustrates the method of automatically updating a web page according to the present invention;

Figure 23 illustrates the method of automatically paging text regions according to the present invention;

Figure 24 illustrates the method of executing Event Triggers according to the present invention; and

Figure 25 illustrates the screen display for the scheduling feature according to thc present invention.

**Detailed Description of thc Invention**

Figure 1 illustrates the operating environment in which the present invention works. Computer system 30 includes computer 32 running an operating system and at least one software application. The operating system can be any known system available on the market. The software application may be a web browser with the ability to execute Java code. Computer 32 includes central processing unit 34 (CPU), at least one input resource 36, at least one output resource 38 and a bus systcm 40. Computer 32 can be a personal computer as is known in the art, a workstation as is known in the art, or another kind of computer.

CPU 34 is known in the art and includes memory 42 comprising main memory 44 and secondary memory 46. Main memory 44 can include Random Access Memory (RAM) and Read Only Memory (ROM). Secondary memory 46 could be ROM, an optical or magnetic disk, or any other type of memory.

Input resource 36 can be a keyboard, mouse, audio device or any other such resource. Output resource 38 can be a display, printer, audio device or any other such resource. The input/output resource can also include a network connection 48 for communication with remote computer systems 50. In the preferred embodiment, the computer is a personal computer running either windows 95, 98 or NT and includes at least a display, keyboard, mouse and network connection. The preferred running software application program is a web browser, preferably either Microsoft Internet

Explorer 5.0® or Netscape Navigator 4.7®. The bus system transmits data between the eomponents.

Figure 2 illustrates the browsing environment of the present invention. Computer system 30 interfaees with eomputer network 48 and eommunieates with remote computer 50. Computer network 48 is preferably an Internet/Intranet connection but can be some other type of connection as well. Computer system 30 connects to network 48 via network interface cireuit 52 which sends data to and reeeives data from remote eomputer system 50. Interface 52 ineludes an interfacc card 54, physical eonnection 56 such as telephone line or Integrated Services Digital Network (ISDN) and connection device 58 sueh as a modem or cable.

As known in the art, the browser is used to gain aeeess to and retrieve data from documents on a remote eomputer. In the preferred embodiment, the doeuments include web sites on the Internet. Web pages are stored as files on the remote eomputer and the Uniform Resouree Locator (URL) speeifies the file location. Commonly, the browser sends requests to a web site, the web server for the web site retrieves the information and sends the web browser the information. The web page format ean inelude HTML, text, images, audio, and video along with many other features. The web page is ereated by a web developer at remote computer 50 and is accessed by a user at computer 30.

Figure 3 illustrates method 100 of formatting and creating a web page aceording to the present invention. In step 110, the layout of the web page is speeified by ereating region types for text, graphies, video, and/or date/time. The regions may include audio advertising, news, weather, sports or financial information. Any number of regions can be speeified. The template screen display 60 on the GUI is illustrated in Figure 4. Template 60 ineludes toolbars 62 providing an easy to use interfaee. The toolbars include buttons for identifying eaeh of the four type of regions available. Eaeh region 61 is ereated by a simple drag and drop operation with a mouse as is well known in the art.

In step 120, each region 61 is assoeiated with a piece of content. The content ean be a file or stream on local eomputer or anywhere on the Internet. For example, one region eould be populated with content from Miami, while another region is populated with content from New York. Content can include audio, weather, sports, news, financial, chat, polling and/or advertising.

In step 130, properties for each region 61 are assigned. Properties can include fonts, shadows, paging rate and refresh rate and others.

In step 140, the web page is published to the Internet. When the developer publishes the web page, a software module is created and processes the developer's template selections. The module dynamically generates the required DHTML, Javascript and Java applet to run the web page. Generally, in a method known in the art, the software module uploads the data to computer 30 and displays the web page in HTML format to the viewer.

Each step in creating and formatting a web page is accompanied by a GUI. The web developer specifies a number of properties for the template. The template properties are used to define properties that are used by the web page as a whole. The template properties include General, Background, Size, Audio, Keywords, Meta Tags, Publish and Schedule.

Figure 5 illustrates the GUI for the template properties 64 with the General Tab selected. The General Tab is used to define the web page title and description and to define a transition used when the user leaves this web page. The transition types are defined by the web browser in use.

Figure 6 illustrates the GUI for the template properties 64 with the Background Tab selected. Each web page can have a background image display. The Background Tab is used to define the background image as well as a background colour.

Figure 7 illustrates GUI 64 but with the Size Tab selected. The Size Tab is used to set the size of the web page.

Figure 8 illustrates GUI 64 but with the Audio Tab selected. Each web page may have an audio file playing in the background. This tab is used to select an audio file to play while viewing thc web page.

Figure 9 illustrates GUI 64 but with the Keywords Tab selected. The Keywords Tab is used to define the search engine keywords for this web page. Each web page can have its own set of keywords. The keywords are used to help search engines, such as AltaVista, Excite, Yahoo, and others, to characterise the contents of the web page.

Figure 10 illustrates GUI 64 but with the Meta Tags Tab selected. The Meta Tags Tab is used to define special properties for a web page, such as Author. Meta Tags are known in the art.

Figure 11 illustrates GUI 64 but with the Publish Tab selected. The Publish Tab is used to define special properties to be used when publishing this web page.

Figure 12 illustrates GUI 64 but with the Schedule Tab selected. Each web page can have a Schedule file associated with it. The schedule file is a XML file that includes a series of times and Event Triggers to execute at the appropriate times. The Schedule feature and Event Triggers will be explained in detail below.

The present invention also allows template 60 to be saved and reused.

In step 110 of method 100, the web developer specifies various regions in the template where content is to appear. The user is also able to specify the properties for each region. Their properties control the appearance of the region and any dynamic effects that may occur within the region. This step is also accompanied by a GUI.

Figure 13 illustrates a GUI for the region properties 66 with the General Tab selected. The General Tab is used to specify an initial piece of content to be displayed in the region and to define a name for the region. Each region can be identified with a unique name. In the case of a date/time region, the format of the date/time is set here as well.

Figure 14 illustrates GUI 6 but with the Paging Tab selected. The Paging Tab is used to control the dynamic paging of a Text or HTML region. The Paging Rate determines the length of time each page remains visible to the viewer. The transition defines the method used to switch between pages and are typical transition choices, such as snap, box in or box out. The Article Defined By and Page Defined By sections are used to determine how articles are defined. Each article comprises a number of pages. Each article may also have a title. The paging feature displays the article title and each page of the article at the defined paging rate. The title of the article remains displayed for every page of that article. When all the pages of the article have been displayed, the next article is displayed along with its title. Automatic Region Update defines how often the software running in the web browser will update the contents of the region. The paging feature will be described in detail below.

Figure 15 illustrates GUI 66 but with the Display Tab selected. The Display Tab shows the X-Y location of the region and the width and height of the region. In addition, the web developer can control whether a small control panel is displayed, or whether the region is hidden. For video regions, the developer can choose to disable the right click option on a mouse, which will limit the actions available to a viewer.

Figure 16 illustrates GUI 66 but with the Text Attributes Tab selected. This tab is used to set all standard text properties for a text/HTML region.

Figure 17 illustrates GUI 66 but with the Text Styles Tab selected. This tab is used to set colour, shadow, and glow properties for the text and HTML regions.

Figure 18 illustrates GUI 66 but with the HTML Attributes Tab selected. This tab is used for graphics regions. The web developer can create links to other web pages using this tab.

Figure 19 illustrates GUI 66 but with the User Events Tab selected. The User Events Tab is used to set rollover images for graphics and to enter Event Triggers that should occur when a Mouse Click, Mouse Over or Mouse Exit occurs.

Figure 20 illustrates GUI 66 but with the Padding Tab selected. The Padding Tab is used to define the distance between the edge of a text box and the text contained within that box.

Figure 21 illustrates a typical web page 86 that can be created with the above described steps. Although web pages can have any appearance, Figure 21 illustrates many elements typically found. The page includes  video region 68, two text regions 70 and 72, a time region 74 and four graphics regions 76, 78, 80, 82 (the background is a large graphic image).

The present invention also allows a published web page created by the above steps to be automatically updated. If content in a region on a web page, for example the one in Figure 21, is changed at the remote computer, the web page region is automatically updated at the viewer's computer without having to republish the web page and without the user having to refresh the page. The web page is automatically updated by method 200 as illustrated in Figure 22.

In step 210, a Java applet in the web browser monitors the source of the content in each template region. In step 220, the browser Java applet determines if the content has changed. If yes, then in step 230, the browser Java applet parses the source code of the updated content. In step 240, the new content is downloaded from the source and reloaded into the web page region automatically. The affected region is updated on the web page without affecting the rest of the page. The web page developer can define the interval for which the browser applet updates the content from the server. This interval can be unique for each region.

A web page created by method 100 can also include text regions that automatically page through contents of the source file. The paging feature is useful when a text region includes content larger than the specified display region. Paging allows the text to be divided into separate pages, and each page can then be displayed in the region periodically. Paging is accomplished according to method 300 illustrated in Figure 23. In step 310, a Java applet in the browser parses the source file into a number of articles. The source file may be an ASCII text file or a complete HTML file. In step 320 the browser determines if each article content is larger than the user specified text region. If so, then in step 330 the article is divided into a number of pages. In step 340, each page is then displayed automatically at a specific rate (Paging Rate) identified by the web developer. If in step 320 it is determined that the content is not larger than the specified region, then step 350 is performed and the content is displayed as is in the region.

The invention can also include a Manual Paging Control for each text or HTML region. The Manual Paging Control allows the web site viewer to take control of a region and manually select the content to be viewed.

As previously mentioned the present invention also provides Event Triggers and Scheduling features.

The invention allows the web page to be enhanced through the use of Event Triggers, which may be used to control the operation and appearance of the web page and may be used to synchronize web page content with video content, for example. Since Event Triggers are built into the Java applet, they are automatically handled by the present invention.

Event Triggers can be generated from a variety of sources including events embedded within the video stream, events caused by user actions such as Mouse Clicks, Mouse Enters, and Mouse Exits, events embedded as commands within the text of an HTML or Text Template Region or events triggered by Time-Of-Day Schedules.

Events can occur simultaneously. For example, a move and a resize can occur at the same time. This will result in a region moving and resizing simultaneously.

An event in one region can trigger an event in another region. For example, a mouse click on a graphic region in a web page may change the video content, and the change in video content may in turn change the text content on the web page.

Each event has a number of parameters. Most events specify a RegionName, which is a string value. Other parameters are either String Values, such as a new source for a region, or Numeric Values such as an X or Y location value.

The Event Triggers supported by the invention include, but are not limited to those listed in Table 1.

| **Event Trigger Type** | **Parameters** | **Action** |
|---|---|---|
| Change Source | Region Name, New Source (file, or URL) | Change the source file for a Region. |
| Change Text | Region Name, Text to be displayed (plain text or HTML) | Change the text in an HTML or Text region to the new text contained in the command parameter. |
| Change Z | Region Name, New Z Order Value | Change the Z Order of the specified Region. Z order is used to determine which display elements appear "in front of" or "behind" one another. |

| Event Trigger Type | Parameters | Action |
|---|---|---|
| Hide | Region Name | Cause a Region to become hidden or invisible. |
| Show | Region Name | Cause a Region to become visible. |
| Move | Region Name, X, Y, Time (in milliseconds) | Cause a Region to move to a new X-Y coordinate location on the web page. This command may also include a time duration over which the move will take place. This allows objects to slide across the screen smoothly. |
| Resize | Region Name, Width, Height, Time (in milliseconds) | Cause a region to change in size to a new width and height. This command may also include a time duration over which the resize will take place. This allows objects to grow or shrink smoothly. |
| Page Flip | | This command is used to open new Browser Windows, and control their content and appearance. |
| Play | Region Name | Cause the content in a Rich Media control such as a video or audio file to begin playing. |
| Pause | Region Name | Cause the content in a Rich Media control such as a video or audio file to pause playing. |
| Stop | Region Name | Cause the content in a Rich Media control such as a video or audio file to stop playing. |
| Rewind | Region Name | Cause the content in a Rich Media control such as a video or audio file to rewind to the beginning. |
| Forward | Region Name | Cause the content in a Rich Media control such as a video or audio file to skip to the next track. |

| Event Trigger Type | Parameters | Action |
|---|---|---|
| Seek | Region Name, Time (seconds) | Cause the content in a Rich Media control such as a video or audio file to seek to a specific location. |
| Schedule Start | Schedule Name | Cause a time-of-day schedule to start running. |
| Schedule Stop | Schedule Name | Cause a time-of-day schedule to stop running. |
| Schedule GoTo | Schedule Name, Time (in milliseconds) | Cause a time-of-day schedule to jump to a specific location within the schedule. |
| Create Region | Region Name, X, Y, W, H, Content Source | Create a new Region in the web page. A number of Region Properties can be set, such as Region Type, Style and Initial Content. |
| Delete Region | Region Name | Permanently delete a region from the web page. |
| Change Region Style | Region Name, Style | Change the Style Properties for a Region to those of the defined Style. |
| Change Propcrty | Region/Style Name, Property, Value | Change the Region Properties for a Region. Properties are generally internal control items such as paging rate, refresh rate, etc) |
| Create Style | Style Name | Create a new Style (styles define a number of visible properties such as colour, font, size, etc.) |
| Create Macro | Macro Name, Command, Propertics | A macro is a collection of Event Triggers that can be referred to and executed by a single name. The Create Macro command parameters include the name of the Macro, and the Event Trigger, and parameters that should be added to the Macro. Create Macro can be called any number of times. Each call with the same Macro Name will add the new event to the macro. |
| Delete Macro | Macro Name | This command is used to clear the contents of a Macro. |

| Event Trigger Type | Parameters | Action |
|---|---|---|
| Schedule Synch | Time, Date | This command is used to set the time used by the applet within the browser for scheduling events. This allows all viewers to experience events at the same time, regardless of small deviations in their computer's clock. |

Table 1: The Event Triggers Supported By Invention

Method 400 of executing the action corresponding to each Event Trigger is illustrated in Figure 24. In step 410, the source file is parsed by the browser Java applet. If in step 420, built in supported Event Triggers in the source file are detected, then in step 430 the browser Java applet automatically executes the corresponding action on the web page when the event occurs. In addition, events will cause the web page to transform in front of the viewer's eyes.

An example of the practical use of Event Triggers is to synchronize web page content such that a HTML region displays a particular advertisement at the same time as an ad plays in a video region.

Scheduling is used to link web page appearance and content to the time of day or to a specific time interval. Web page developers can schedule any of the events defined above to occur at any time during a schedule. The scheduling feature is also accompanied by GUI 84 as illustrated in Figure 25. The GUI allows the developer to create a time line schedule for each region of the web page. A schedule file in an XML format is created from the information provided to the GUI. The schedule file contains one or more schedules to be executed by a web page.

Each schedule contains one or more Events and each Event contains a Message and a Time Value. Each Message includes one of the Event Trigger Types and one or more parameters that are used to control the Event Trigger. Each Time Value has a Type and a Value. The Type can be either RealTime or TimeOfDay. The Time Value for RealTime Events is a number in seconds. The Time Value for TimeOfDay Events consists of a number of elements. One or more of these elements must be present: YEAR, MONTH, DAY, HOUR, MINUTE, SECOND, GMT.

Time Values are calculated based on the time elements that are present. For instance, if only the month value is specified, the event will occur once, whenever the web page is viewed during the matching month. If both the month and the day are specified, the event will occur on that day, regardless of the year. Alternately, if only a time value is specified (for example: HOUR = 15, MINUTE=10), then that event will occur every day at 3:10pm.

Schedules may be defined in one of three time frames. The first is Local Time wherein the Schedule is run based on the time of day at the user's browser. Each user will experience events according to their local time. Users in different time zones will not experience the same event simultaneously. The second is GMT Time wherein the schedule is run based on Grecnwich Mean Time. All viewers of the web site will experience events in a GMT schedule simultaneously, regardless of their respective time zones. The third is Offset Time wherein the schedule is run based on a 0 based start time. This is used when you want a schedule to start a specified length of time after a specific occurrence. For example, the web developer could schedule an event to occur four minutes the web page is opened.

Any number of schedules can be running simultaneously. Schedules can start or stop one another or can be started and stopped by Event Triggers that occur as defined in the four Event Triggers sources. Also, a schedule can start automatically when the web page is loaded.

The schedule file created is preferably in XML according to the following format:

```
<SCHEDULES>
    <MAIN>
        <EVENT>
            <MESSAGE>
                <TYPE>sofTV_ScheduleStart</TYPE>
                <SCHEDULE>SOFTVWEBPAGE</SCHEDULE>
        </MESSAGE>
            <TIME>
                <TYPE>RealTime</TYPE>
                <VALUE>0</VALUE>
            </TIME>
        </EVENT>
        <EVENT>
            <MESSAGE>
                <TYPE>sofTV_Hide</TYPE>
                <REGION>TextRegion2</REGION>
            </MESSAGE>
            <TIME>
                <TYPE>RealTime</TYPE>
                <VALUE>0</VALUE>
            </TIME>
        </EVENT>
    </MAIN>
    <SOFTVWEBPAGE>
        <EVENT>
            <MESSAGE>
                <TYPE>sofTV_PageFlip</TYPE>
                <TARGET>SofTVnet</TARGET>
                <URL>http://www.softv.net</URL>
            </MESSAGE>
            <TIME>
            <TYPE>RealTime</TYPE>
            <VALUE>55000</VALUE>
            </TIME>
        </EVENT>
    </SOFTVWEBPAGE>
</SCHEDULES>
```

Schedules allow a web developer to schedule web page events in advance and have web pages automatically change or update anytime.

In operation, the web developer utilizes the inventive software application incorporating the features described above to create and format a web page at remote computer 50. A Java applet software module is created by the developer when the page is published. Computer 30 then retrieves the web page and the Java applet software module from the remote computer. Computer 30 then can view the web page. The browser on the remote computer executes Java applets to parse the source code to determine and execute updates, paging, Event Triggers and Scheduling. In this way each viewer views, possibly, a different personalized web page.

The inventive software application is built using the C++ programming language. When a web page is designed using the application, it is first stored locally on the web designer's computer as an XML formatted file that defines the layout of the web page and the properties of the page itself and all the regions contained on the page.

When a user selects the publish function, the application automatically creates the HTML, DHTML and JavaScript codes required to load and run the web page in the viewer's browser. Then the web page is saved in a directory chosen by the user. Additionally, the application copies any required Java applets, and all required content items into the publish directory. The Java Applets include, but are not limited to:

- DateTimeRegion.js (manages the formatting of the date and time region)
- General.js, (General utilities shared by the Java Applets)
- ImageRegion.js, (manages the display of graphics in an Image Region)
- Messaging.js, (manages the parsing and execution of Event Triggers)
- Scheduling.js, (manages the execution of a schedule)
- TextRegion.js, (manages the parsing and display of text and HTML regions)
- VideoRegion.js, (manages the display of video content)
- XMLParser.js, (parses XML formatted files into internal data structures)
- net.softv.zip. (general utilities used to managc web pages created with the invention)

The Java applets run in the browsers of every viewer of the web page. The applets are started by the JavaScript on the web page, and are used to actually perform the mechanics of running the web page. In the case of an HTML region, the applets:

- load the content into the web page,
- parse the content files into articles,
- detect, and execute any Event Triggers embedded within the text of an Article,
- display the individual pages of the articles into the specified region of the web page
- manage the timed display of each page of the article,
- manage the refreshing of the article from the source content file,
- provide formatting functions such as creating a shadow effect for the text.

In the case of video regions, the Java Applets control the video playback application, which can be a Windows Media Technology Player, a Real Networks Player, or an Apple Quicktime Player.

In the case of a Graphics region, the Java Applet loads the graphic image into the region, resizing the graphic if required. The applet also monitors the original source file for the graphic, and if a change is detected, reloads the graphic automatically.

In the case of a date / time region, the Java Applet obtains the current time from the computer's internal clock, then formats the time based on the time specification chosen for the time region. The time can be formatted in a number of different styles, including any combination of hours, minutes, seconds, day, month, year, in either numeric (1,2,3...) or English word (Jan, January, Mon, Monday, etc...) format.

The Java Applets also manage the execution of any Trigger Events that are executed on a web page. The JavaScript on the web page detects the Trigger Event in a video stream, then calls the Java Applet to parse the event parameters, and actually execute the event.

Finally, the Java Applets manage the execution of any schedules attached to the web page. The schedule file is loaded, then converted from XML into an internal data structure. The Java Applets monitor the time, then run the required events in the schedule file at the appropriate time.

The invention may also include a computer readable medium containing instructions executable by the computer for performing any of the features detailed above, or any combination of features thereof.

While the invention has been described according to what is presently considered to be the most practical and preferred embodiments, it must be understood that the invention is not limited to the disclosed embodiments. Those ordinarily skilled in the art will understand that various modifications and equivalent functions may be made without departing from the spirit and scope of the invention as defined in the claims. Therefore, the invention as defined in the claims must be accorded the broadest possible interpretation so as to encompass all such modifications and equivalent functions.

WHAT IS CLAIMED IS:

1. A method of creating a web page comprising the steps of:

specifying regions in a web page template;

associating each region with content; and

assigning properties for each region.

2. An apparatus for creating a web page, the apparatus comprising:

means for specifying regions in a web page template;

means for associating each region with content; and

means for assigning properties for each region.

3. A computer readable medium containing code for creating a web page comprising:

code for specifying regions in a web page template;

code for associating each region with content; and

code for assigning properties for cach region.

4. A method of creating a web page as claimed in claim 1, wherein each region is specified to be either video, graphic, text or date and time.

5. A method of creating a web page as claimed in claim 1, wherein the web page automatically updates web page regions without the use of manual programming comprising the steps of:

monitoring for updated information in the source file;

parsing the updated source file; and

automatically adding the updated information to the web page.

6. A method of creating a web page as claimed in claim 1, wherein text regions are adapted to automatically page without the use of manual programming in a web page comprising the steps of:

parsing the source file for the web page region;

determining from the source file if the content is larger than the display area for the content; and

if yes, then dividing the content into pages; and displaying each page in the display area sequentially, each page being displayed for a specific length of time.

7. A method of creating a web page as claimed in claim 1, wherein the method further comprises the steps of :

identifying an event trigger, the event trigger being built in the instructions whereby event triggers can be identified without programming by the user; and

executing the event trigger.

8. A method of creating a web page as claimed in claim 7, wherein the method further comprises the steps of :

creating a schedule of at least one event trigger,

the schedule being created with use of a graphical user interface whereby the schedule can be created without programming.

9. A method of creating a web page as claimed in claim 1, wherein the text content of the web page is synchronized with the video content of the web page without the use of manual programming, including the steps of:

parsing the source file;

detecting event triggers embedding in video; and

updating the text content according to the detected event trigger.

10. A method of creating a web page as claimed in claim 8 wherein the event triggers the execution of the schedule.

11. A method of creating a web page as claimed in claim 10, the web page including time of day scheduling defined by a time value and a message.

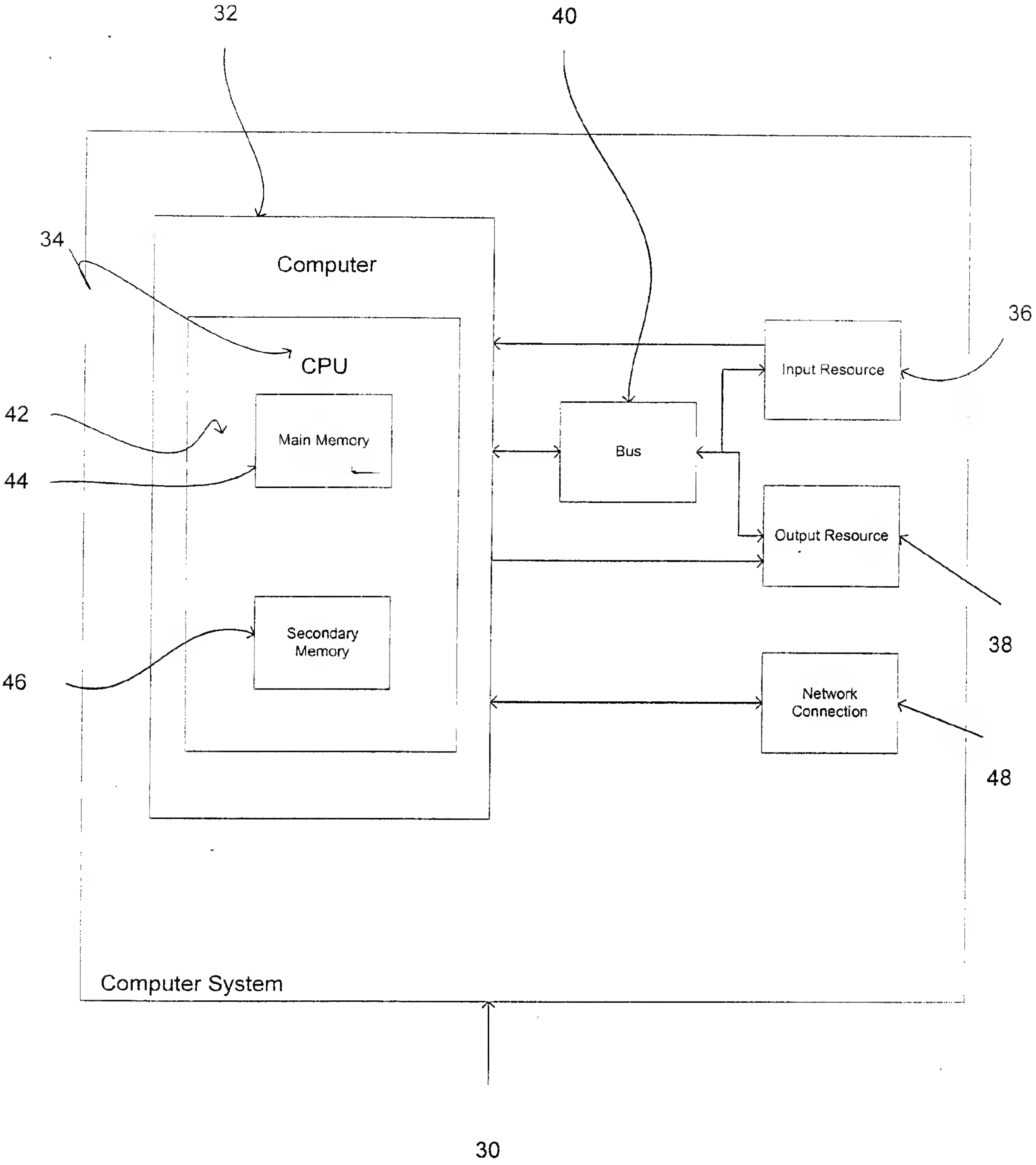

32
40
34
Computer
CPU
36
Input Resource
42
Main Memory
Bus
44
Output Resource
38
Secondary Memory
46
Network Connection
48
Computer System
30

Figure 1

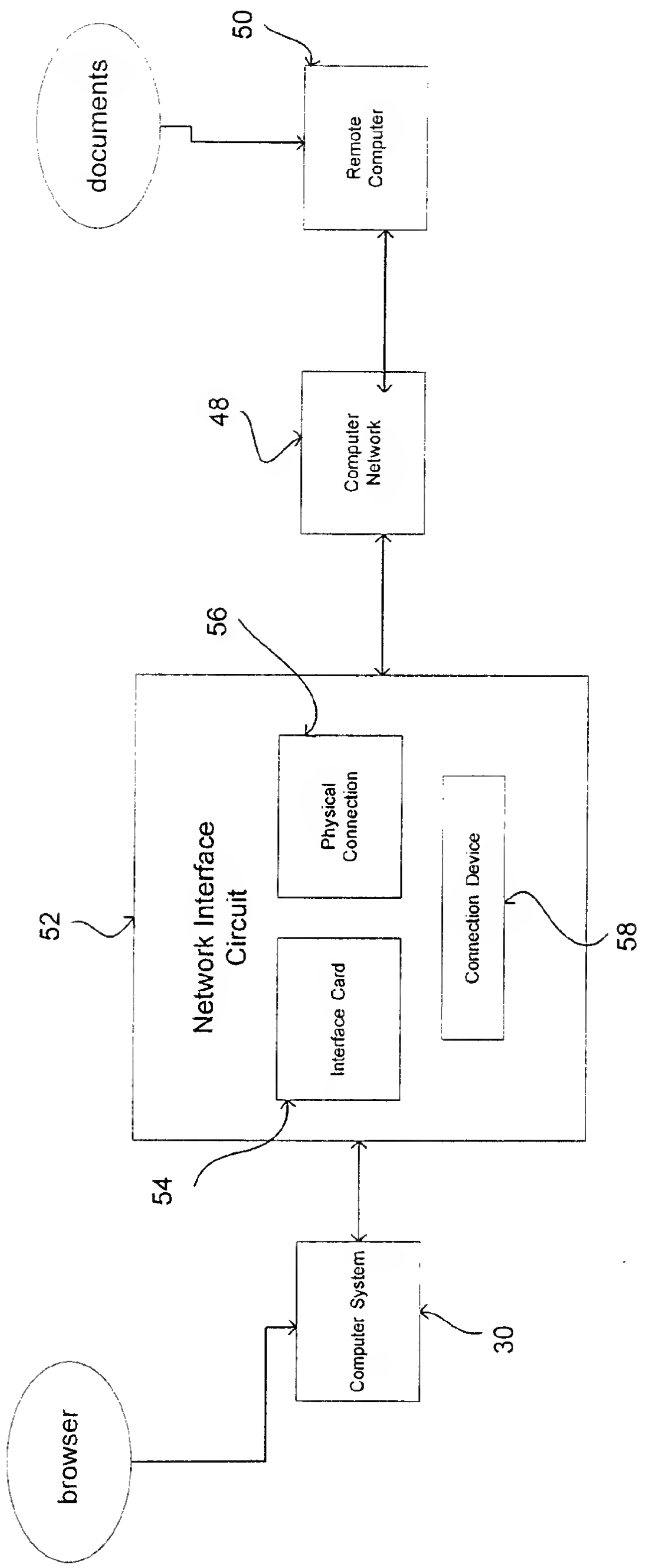

documents
50
Remote Computer
48
Computer Network
56
Network Interface Circuit
52
Physical Connection
Interface Card
Connection Device
58
54
Computer System
30
browser

Figure 2

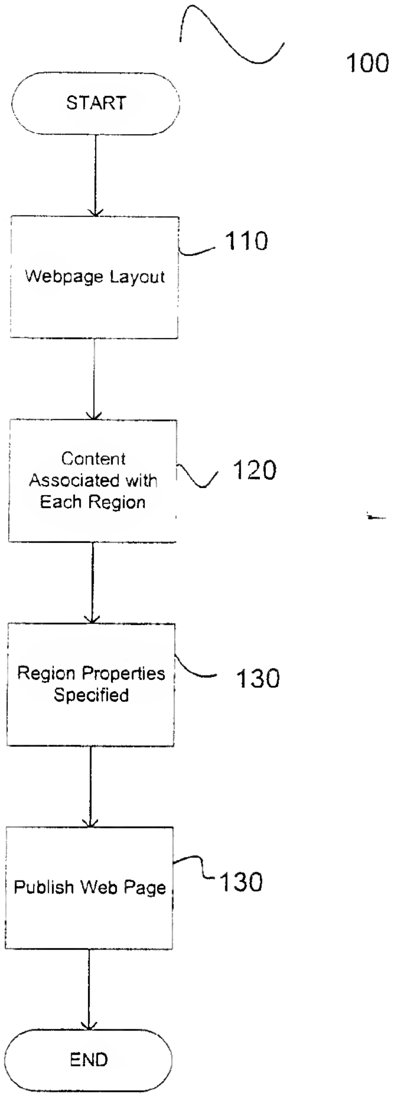

100
START
Webpage Layout
110
Content Associated with Each Region
120
Region Properties Specified
130
Publish Web Page
130
END

Figure 3

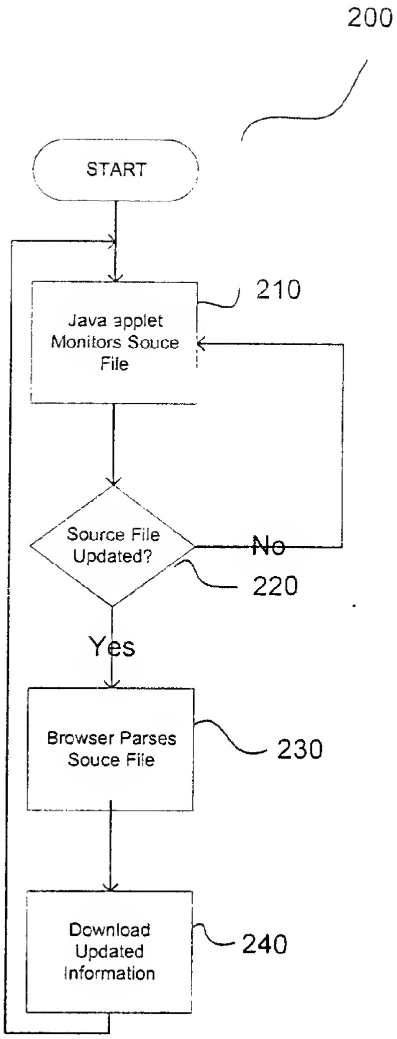

200
START
Java applet Monitors Souce File
210
Source File Updated?
No
220
Yes
Browser Parses Souce File
230
Download Updated Information
240

Figure 22

Gowling, Strathy & Henderson

Figure 4

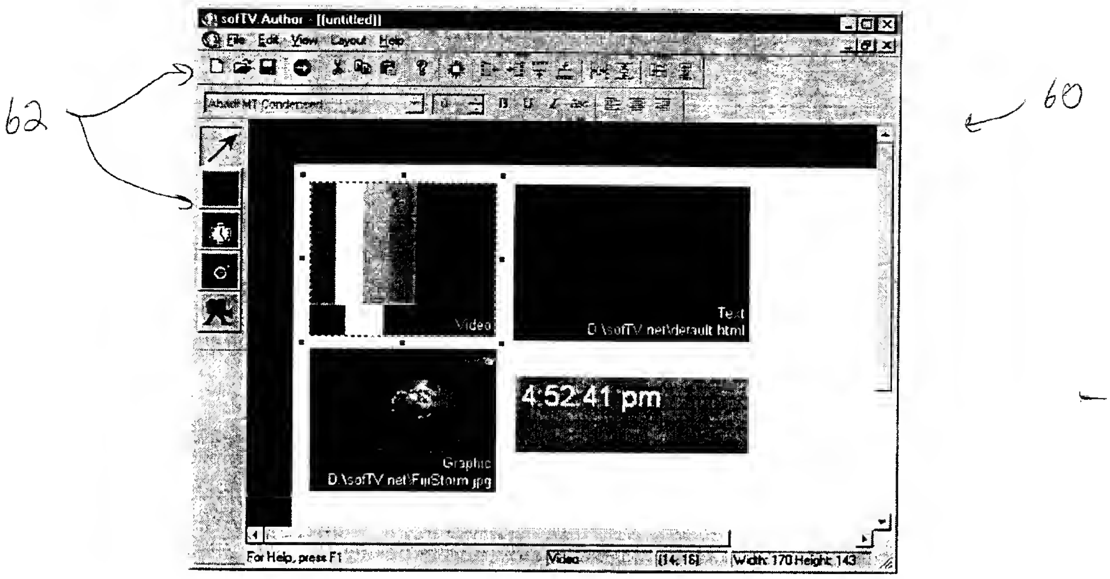
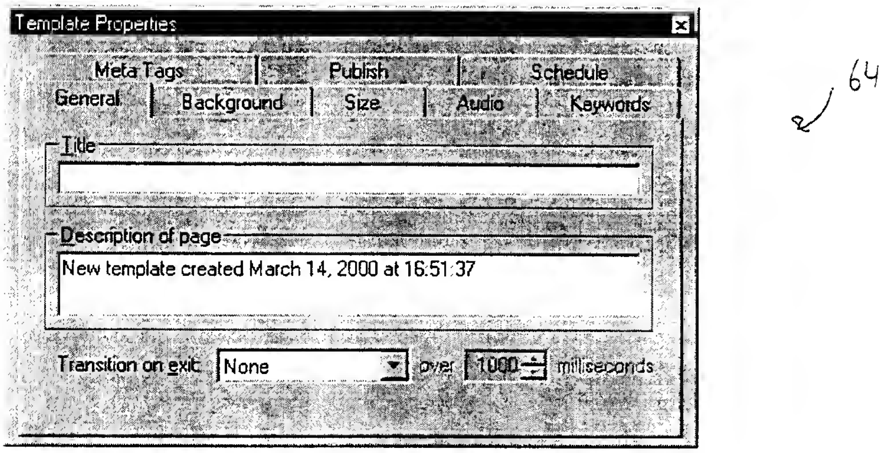

sofTV Author - [[untitled]]
File Edit View Layout Help
Abadi MT Condensed
62
60
Video
Text
D:\sofTV.net\default.html
4:52:41 pm
Graphic
D:\sofTV.net\FinStorm.jpg
For Help, press F1
Video
(14, 16)
Width: 170 Height: 143

Figure 5

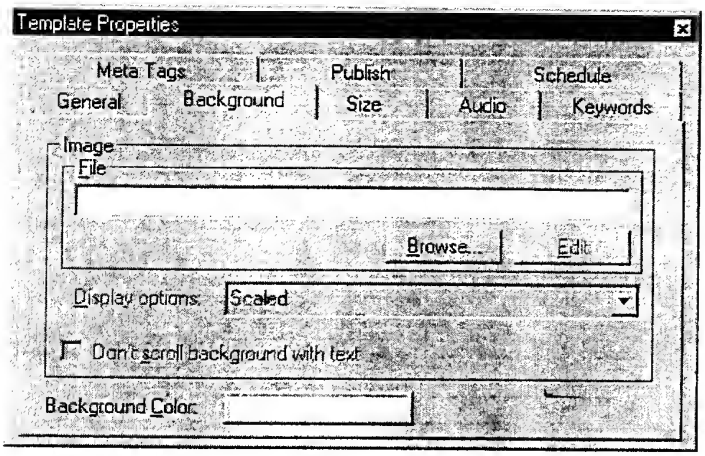


Title:

Description of page: New template created March 14, 2000 at 16:51:37

Transition on exit: None over 1000 milliseconds

## Figure 6



Template Properties

Meta Tags | Publish | Schedule

General | Background | Size | Audio | Keywords

Image

File

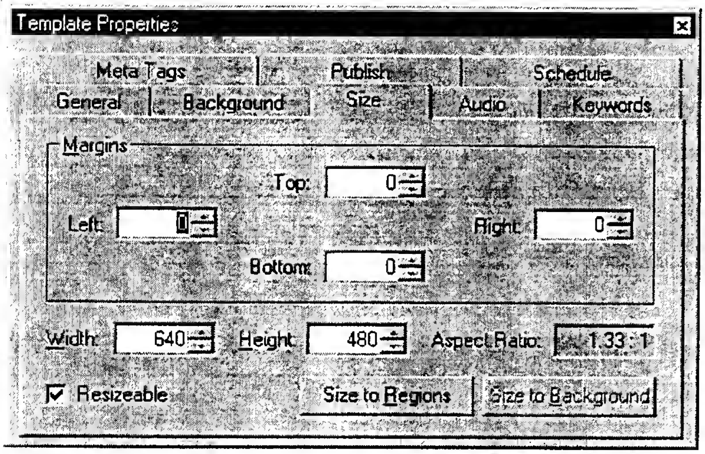
Browse... | Edit

Display options: Scaled

□ Don't scroll background with text

Background Color:

## Figure 7



Template Properties

Meta Tags | Publish | Schedule

General | Background | Size | Audio | Keywords

Margins

Top: 0

Left: 0

Right: 0

Bottom: 0

Width: 640 | Height: 480 | Aspect Ratio: 1.33 : 1

☑ Resizeable

Size to Regions | Size to Background

Figure 8

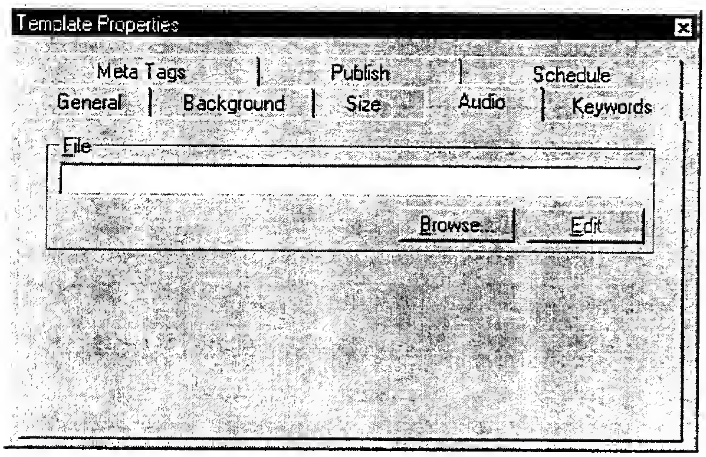
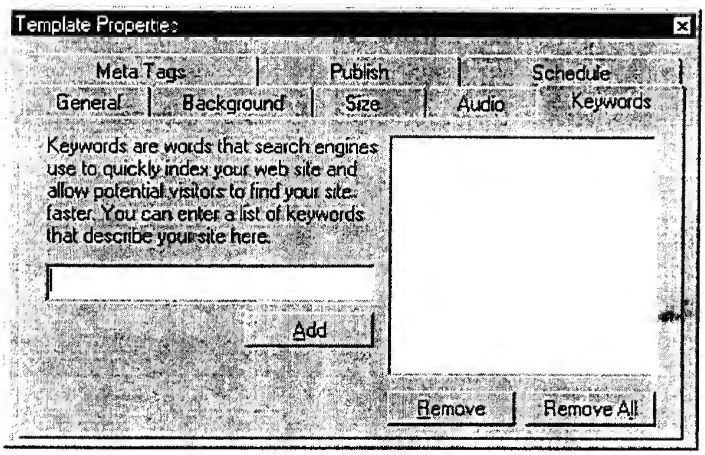
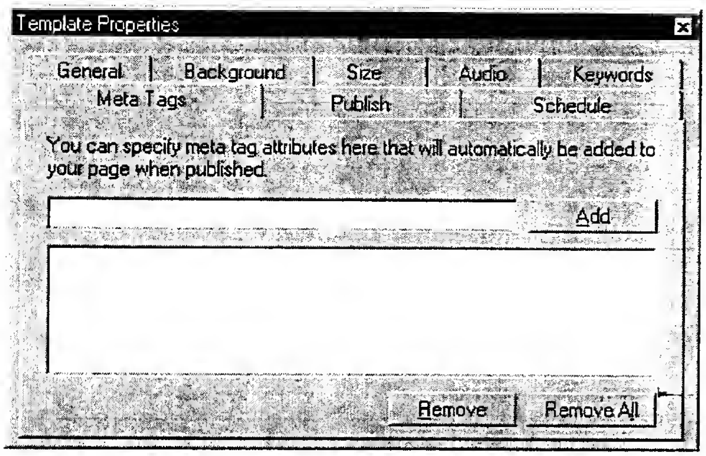


Figure 9



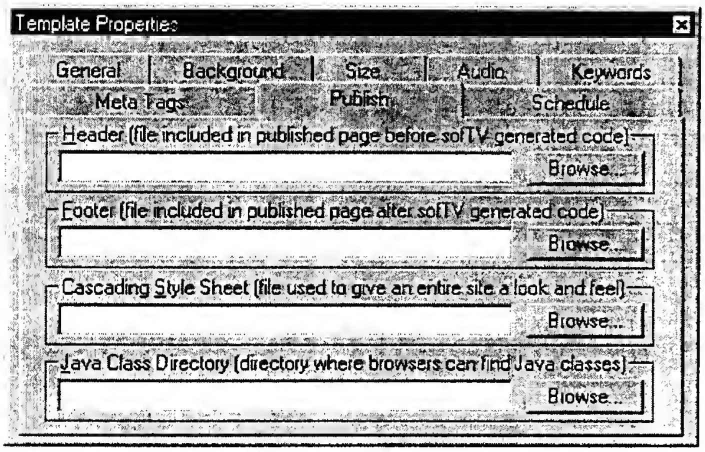
Keywords are words that search engines use to quickly index your web site and allow potential visitors to find your site faster. You can enter a list of keywords that describe your site here.

Figure 10



You can specify meta tag attributes here that will automatically be added to your page when published.

Figure 11



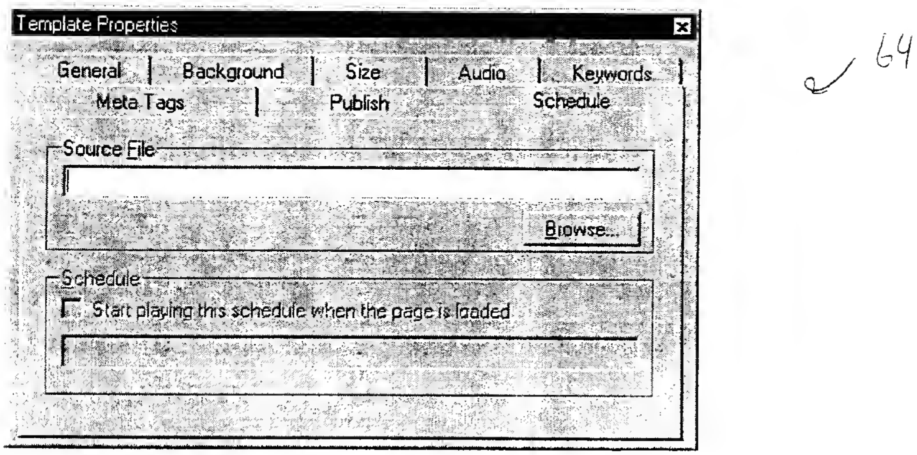
Header (file included in published page before sofTV generated code)

Footer (file included in published page after sofTV generated code)

Cascading Style Sheet (file used to give an entire site a look and feel)

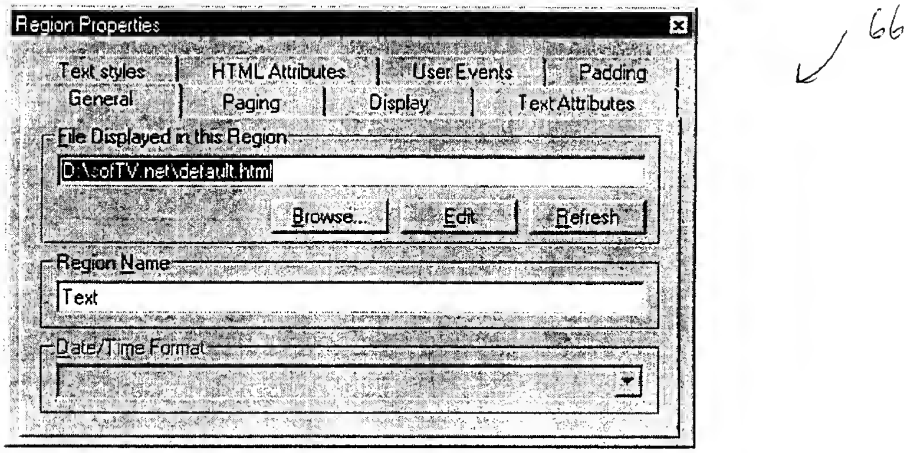
Java Class Directory (directory where browsers can find Java classes)

Figure 12



□ Start playing this schedule when the page is loaded

Figure 13

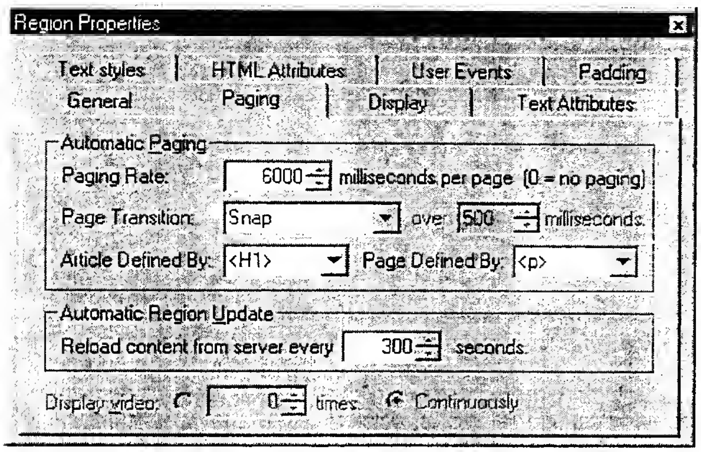


File Displayed in this Region: D:\sofTV.net\default.html

Region Name: Text

Date/Time Format:

## Figure 14



Automatic Paging

Paging Rate: 6000 milliseconds per page (0 = no paging)

Page Transition: Snap over 500 milliseconds

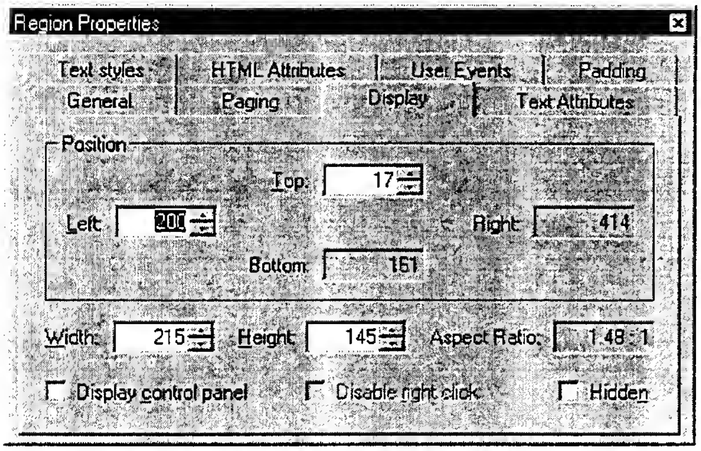
Article Defined By: <H1> Page Defined By: <p>

Automatic Region Update

Reload content from server every 300 seconds

Display video: ○ 0 times ● Continuously

## Figure 15



Position

Top: 17

Left: 200

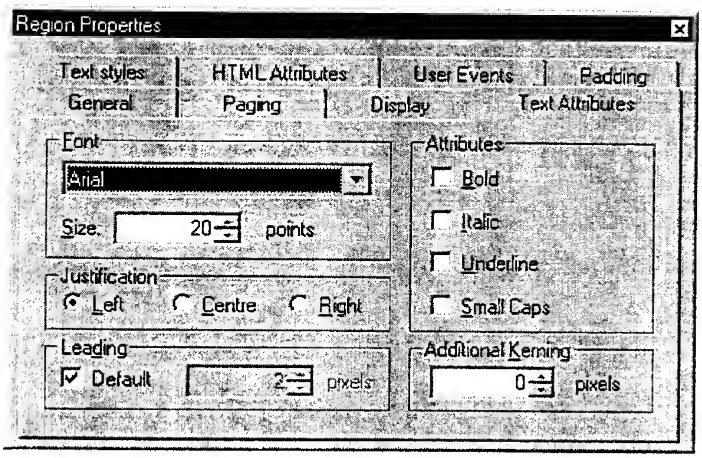
Right: 414

Bottom: 161

Width: 215 Height: 145 Aspect Ratio: 1.48 : 1

□ Display control panel □ Disable right click □ Hidden

Figure 16



Font
Arial
Size: 20 points

Justification
● Left ○ Centre ○ Right

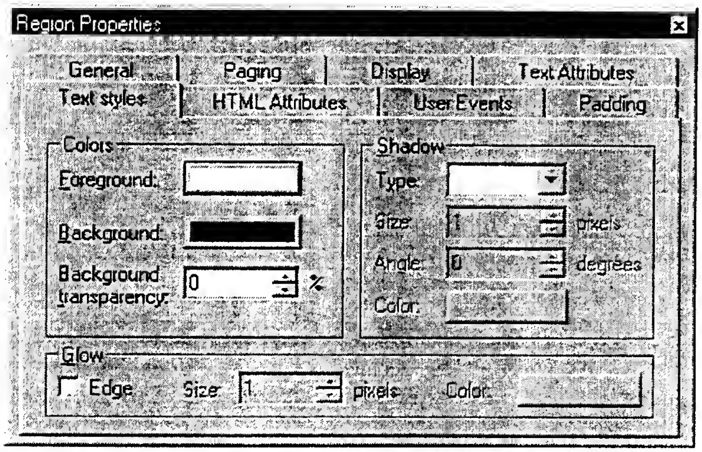
Leading
☑ Default 2 pixels

Attributes
□ Bold
□ Italic
□ Underline
□ Small Caps

Additional Kerning
0 pixels

66

Figure 17



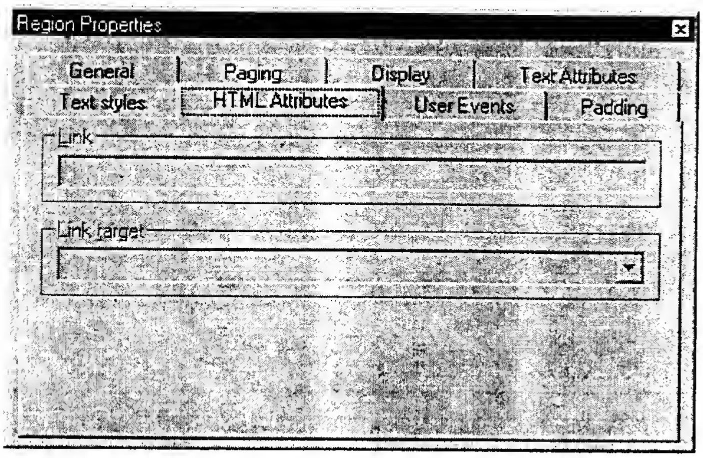
Colors
Foreground: □
Background: ■
Background transparency: 0 %

Shadow
Type:
Size: 1 pixels
Angle: 0 degrees
Color:

Glow
□ Edge Size 1 pixels Color

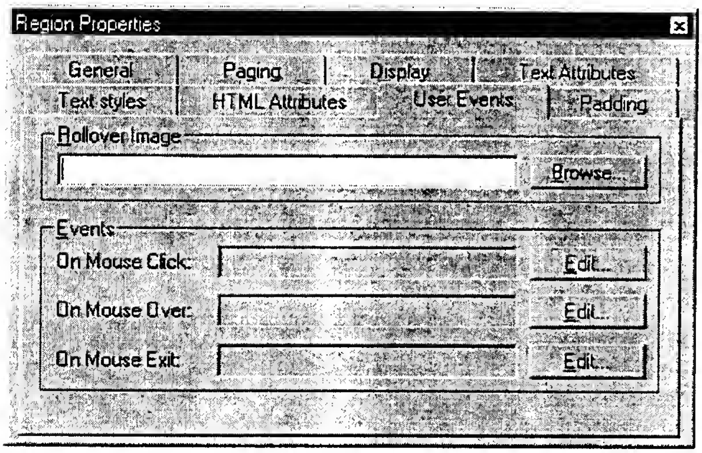
66

## Figure 18



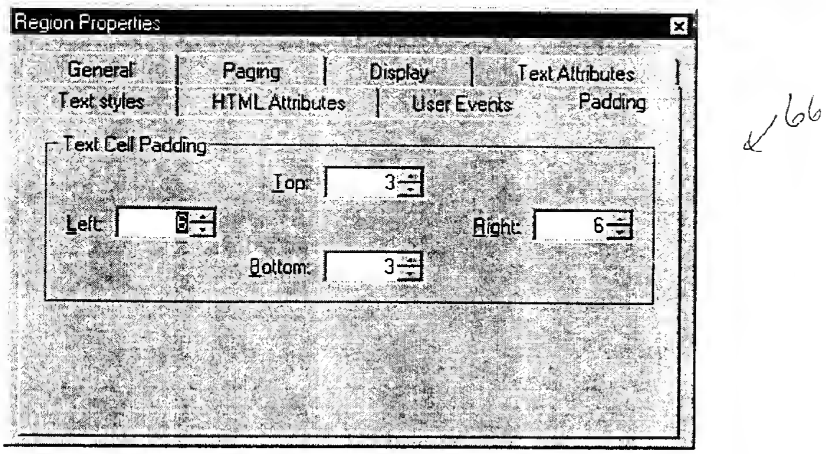
✓ 66

## Figure 19



✓ 66

Figure 20



Text Cell Padding

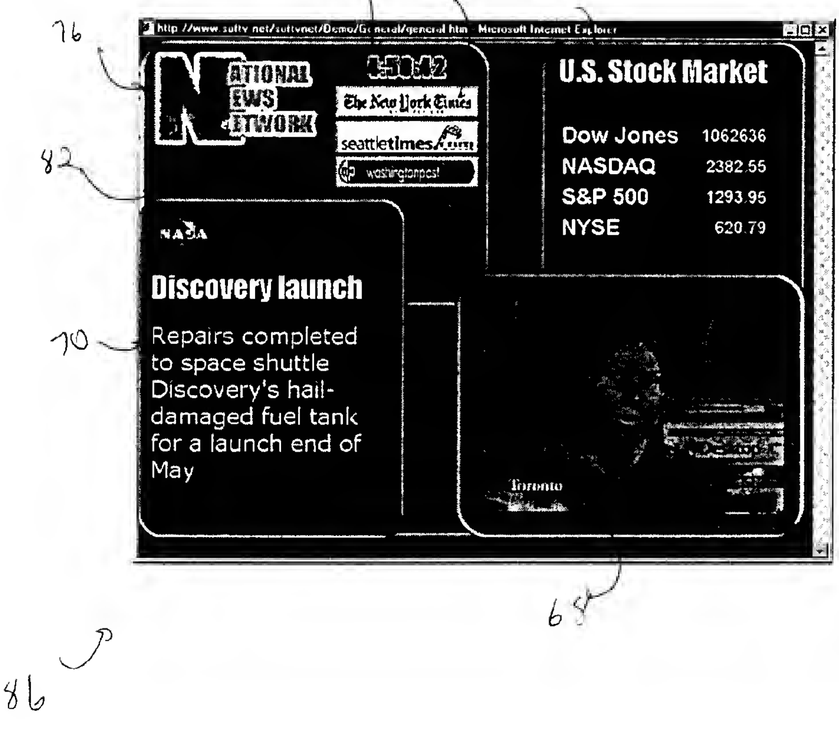
Top: 3

Left: 6

Right: 6

Bottom: 3

Figure 21



U.S. Stock Market

| | |
|---|---|
| Dow Jones | 1062636 |
| NASDAQ | 2382.55 |
| S&P 500 | 1293.95 |
| NYSE | 620.79 |

Discovery launch

Repairs completed to space shuttle Discovery's hail-damaged fuel tank for a launch end of May

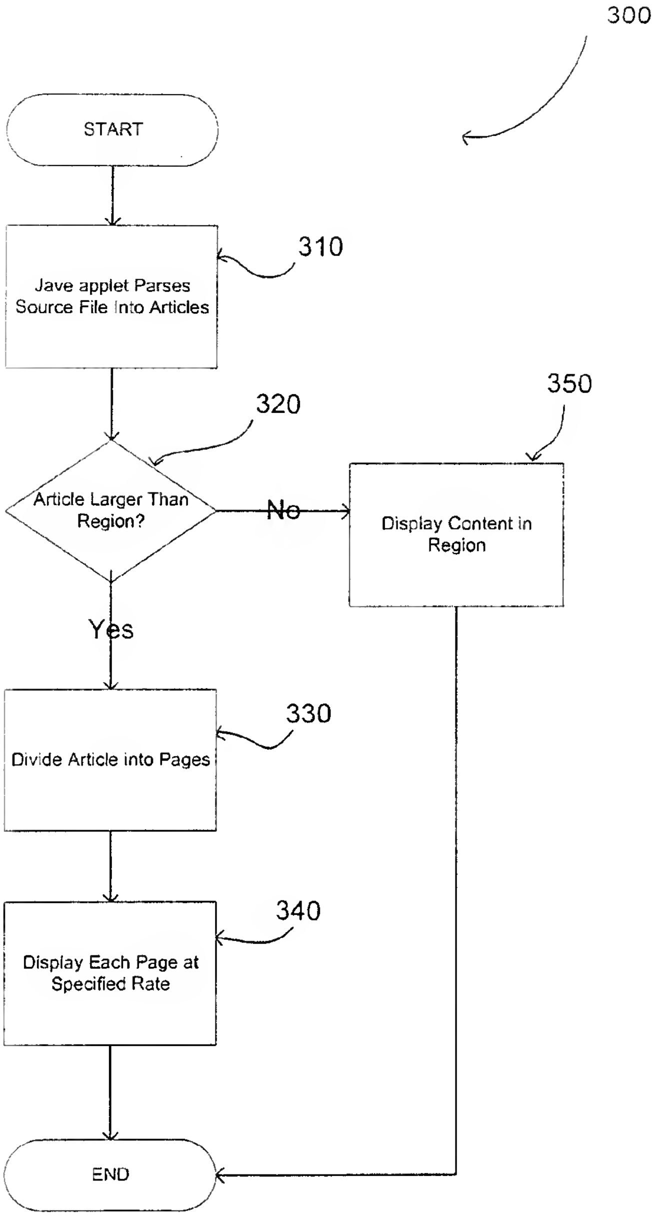

300
START
310
Jave applet Parses Source File Into Articles
320
350
Article Larger Than Region?
No
Display Content in Region
Yes
330
Divide Article into Pages
340
Display Each Page at Specified Rate
END

Figure 23

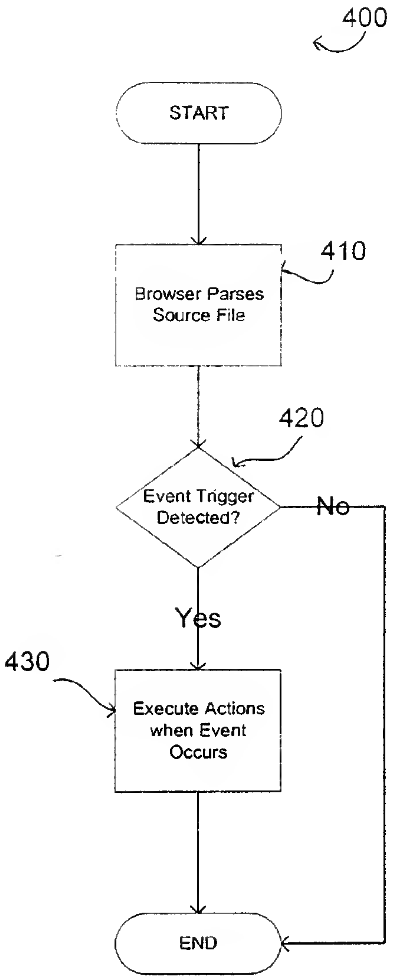
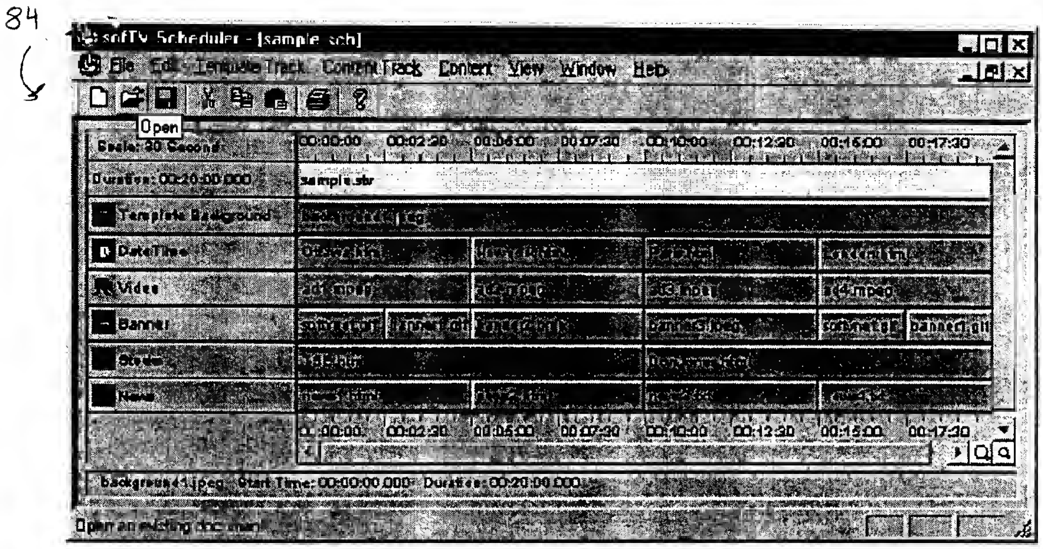

400
START
410
Browser Parses Source File
420
Event Trigger Detected?
No
Yes
430
Execute Actions when Event Occurs
END

Figure 24



background1.jpeg Start Time: 00:00:00.000 Duration: 00:20:00.000

Open an existing document

Figure 25

Gowling, Strathy & Henderson